Bible de projet C++

Table des matières

[**Règles syntaxiques** 2](#_Toc180259431)

[Générale 2](#_Toc180259432)

[Variables 2](#_Toc180259433)

[Fonctions et Méthode 2](#_Toc180259434)

[Classes 2](#_Toc180259435)

[Variables de classe 2](#_Toc180259436)

[Guide line pour Chat GPT 2](#_Toc180259437)

# **Règles syntaxiques**

## Générale

Tous nom doit être en français et minimiser l’utilisation d’abréviation pour maximiser la clarté du code

## Variables

**V[type]\_[nom\_de\_variable]**

Exemple : **Vf\_resulta\_add** serais une variable **flotant** nommé **resulta\_add** en minuscule et des \_ pour séparer les mots du nom

## Fonctions et Méthode

**F[type]\_[NomDeFonction]**

Exemple : **Ff\_Add2Var(float Vf\_a, float Vf\_b)** serais une fonction dont la sortie est un **flotant** nommé **Add2Var** en minuscule et des majuscules pour séparer les mots du nom

## Classes

**C\_[nom\_de\_classe]**

Exemple : **C\_cercle\_colore** serais une classe nommé **cercle\_colore** en minuscule et des \_ pour séparer les mots du nom

## Variables de classe

**m[type]\_[nom\_de\_variable]**

Exemple : **mf\_resulta\_add** serais une variable **flotant** nommé **resulta\_add** en minuscule et des \_ pour séparer les mots du nom

# Guide line pour Chat GPT

L’utilisation de Chat GPT est acceptable mais vous trouverez une liste de conseille pour éviter le temps perdu :

1. Diviser le code en petite portion pour éviter les erreurs
2. Vérification fréquente de la fonctionnalité du code
3. Lui donner les règles syntaxiques pour éviter de repasser sur les noms